第5章 数据卷

1 数据卷概述

数据卷简介

数据卷是一种特殊的目录关系，它将宿主机目录与容器中的目录进行了直接关联，在任何一端目录中的文件写操作，就是对数据卷目录的操作，在另一端都会发生相应变化。相互关联的宿主机端目录与容器端目录，我们一般称为数据卷的挂载点。

数据卷的设计目的就是为了实现数据持久化。数据卷完全独立于容器的生命周期。因此，容器删除时，不会删除其挂载的数据卷。

数据卷的特性

- 数据卷在容器启动时初始化，如果容器使用的镜像在数据卷挂载点本身就包含了数据，那么，这些数据会在容器启动过程中拷贝到新初始化的数据卷中。也就是说，宿主机端的数据卷挂载点目录中也就出现了这些数据
- 数据卷可以在容器之间共享和重用
- 可以对数据卷里的内容直接修改，修改后马上生效，无论是容器内操作还是本地操作
- 数据卷会一直存在，即使挂载数据卷的容器已经被删除

2 创建读写数据卷

读写数据卷指的是容器对挂载点的文件具有读写权限。数据卷是在使用docker run启动容器时指定的，创建具有读写权限的数据卷的语法格式为：

docker run –it –v /宿主机目录绝对路径:/容器内目录绝对路径 镜像

> *注：无论是宿主机还是容器，如果指定的挂载点目录不存在，docker都会自动创建。*

3 创建只读数据卷

只读数据卷，指的是容器对挂载点的操作权限是只读的。宿主机操作权限始终是读写的。因为有些情况下，为了防止容器在运行过程中对文件产生修改，就需要创建只读数据卷。

该命令仅比之前的命令仅多了:ro，具体语法如下：

docker run –it –v /宿主机目录绝对路径:/容器内目录绝对路径:ro 镜像

4 数据卷共享

当一个容器与另一个容器使用相同的数据卷，即不仅共享了宿主机中的挂载点目录，同时还具有相同的容器挂载点目录。此时就称这两个容器实现了“数据卷共享”。数据卷容器是实现数据卷共享的一种非常有效的方案。

当一个容器C启动运行时创建并挂载了数据卷，若其它容器也需要共享该容器C的数据卷，此时容器C就称为数据卷容器。其它容器在docker run启动时只需要通过--volumes-from[容器C] 选项即可实现数据卷共享。

5 安装MySQL

为了保证数据的安全性，在生产环境下安装的mysql容器，在启动时都会使用数据卷来持久化数据。

启动MySQL容器

```
docker run --name mysql \
-e MYSQL_ROOT_PASSROD=111 \
-v /root/mysql/log:/var/log/mysql \
-v /root/mysql/data:/var/lib/mysql \
-v /root/mysql/conf:/etc/mysql/conf.d \
-dp 3306:3306 mysql:5.7
```

新建my.cnf

在宿主机的/root/mysql/conf目录中新建my.cnf文件，并在其中键入如下内容：

```
[client]
default_character_set=utf8

[mysql]
default_character_set=utf8

[mysqld]
character_set_server=utf8
```

重启mysql容器

由于修改了mysql配置，所以需要重启mysql容器，以使新配置生效。

6 安装Redis

创建挂载点目录

首先要在宿主机/root目录中创建一个目录redis，将来用于存放外挂文件redis.conf。

复制redis.conf文件

上传一份redis核心配置文件redis.conf到宿主机目录/root/redis中。

修改redis.conf

修改redis.conf配置文件，将IP绑定解除、关闭保护模式、关闭守护模式。如果需要更换持久化目录，则需要修改dir属性。

启动Redis容器

```
docker run --name myredis \
-v /root/redis/redis.conf:/etc/redis/redis.conf \
-v /root/redis/data:/data \
-dp 6379:6379 redis:7.0 \
redis-server /etc/redis/redis.conf

```

7 Docker网络

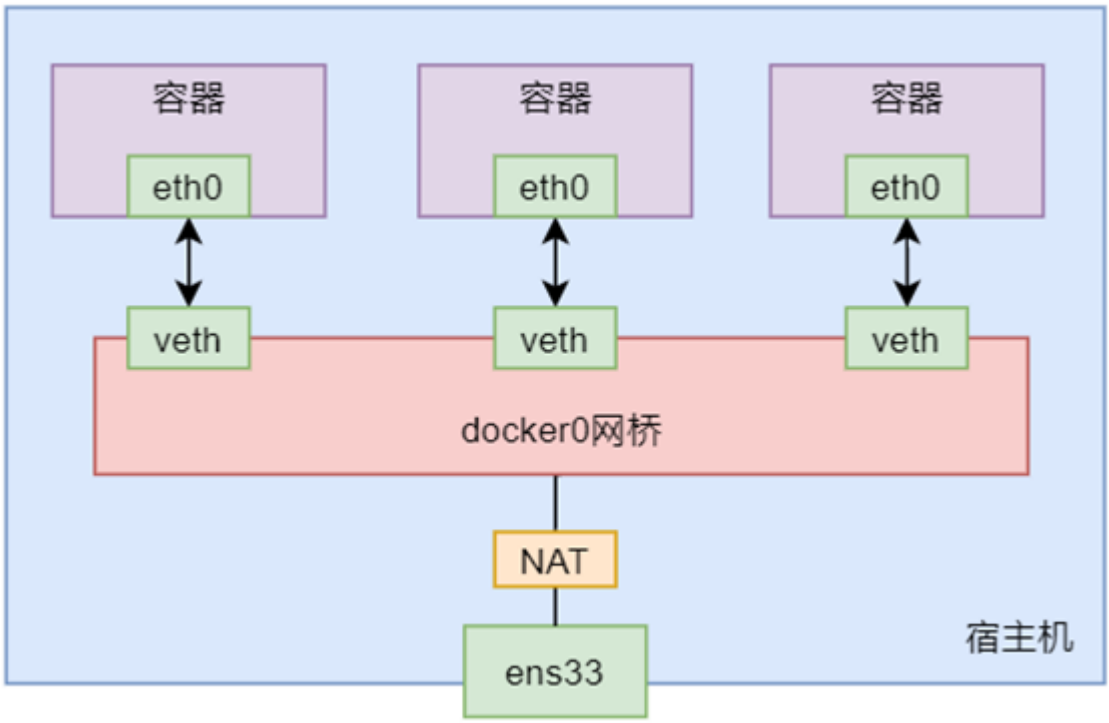
在Docker的宿主机中容器间是互通的，容器与宿主机间也是互通的。而这个互通的实现是通过Docker网络完成的。

通过docker network ls命令可以查看到当前Docker主机中存在的网络列表。有三类网络：bridge、host与none。

在docker run命令中通过--network [网络name]可以指定当前要启动的容器要连接到的网络。

bridge网络

通过docker network inspect bridge可以查看到当前bridge网络中已经连接的容器，及其它相关详情。在Linux主机上，Docker的bridge网络由Bridge驱动创建，其在创建时会创建一个默认的网桥docker0。



host网络

host网络，即与宿主机host共用一个Network Namespace。该网络类型的容器没有独立的网络空间，没有独立的IP，全部与host共用。

none网络

none网络，即没有网络。容器没有网络接口，没有IP。

网络创建

通过docker network create命令可以创建指定名称与类型的网络。默认创建的是bridge网络，通过-d选项可以指定要创建的网络类型。